

Discovery Initiative



Conduire OpenStack Vers l'Edge Computing

Anthony Simonet

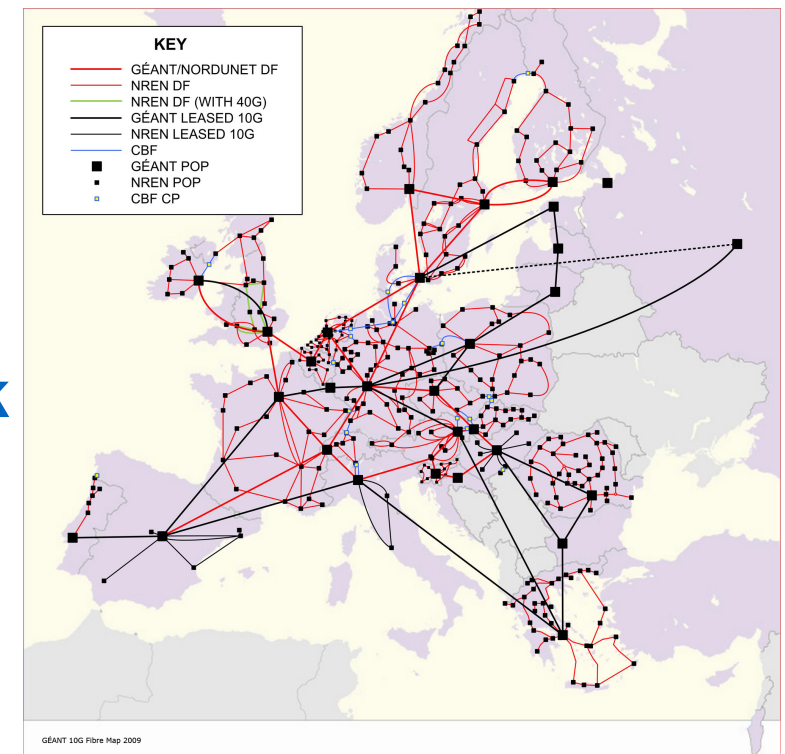
Inria, École des Mines de Nantes, France

Fog/Edge Computing Infrastructures

- Leverage network backbones

Extend any point of presence of network backbones (aka PoP) with servers (from network hubs up to major DSLAMs that are operated by telecom companies, network institutions...).

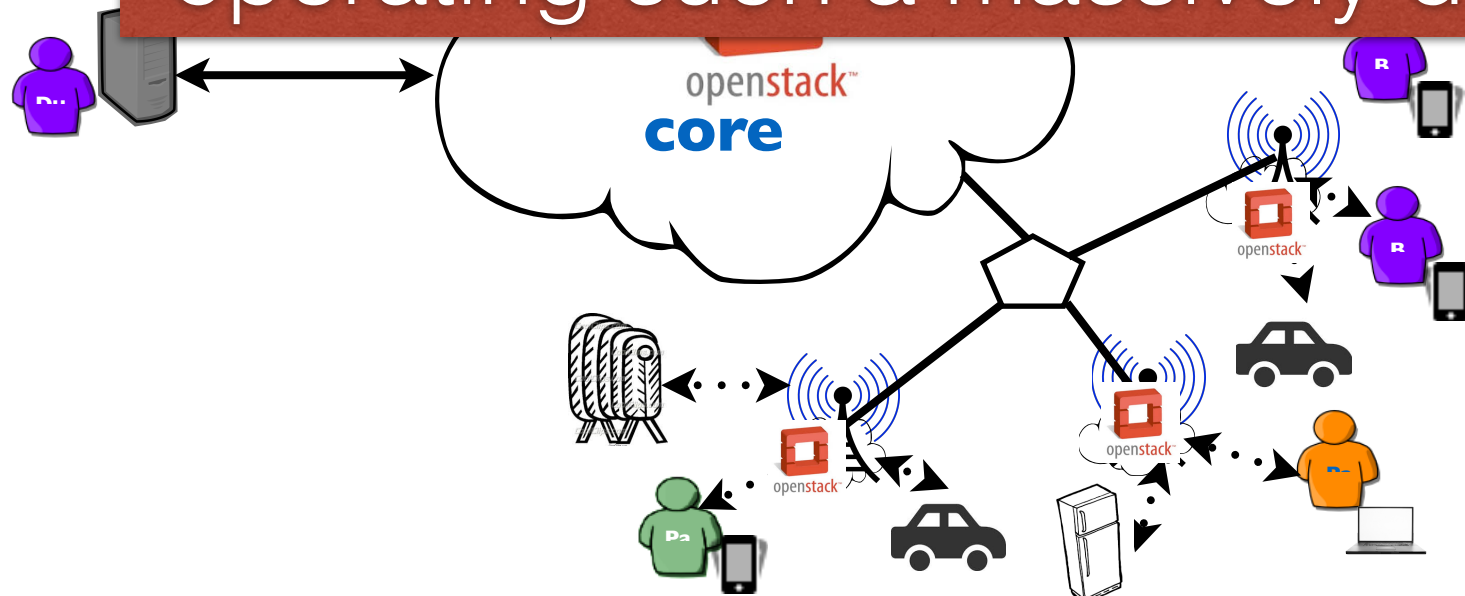
- Extend to the edge by including wireless backbones



European NREN

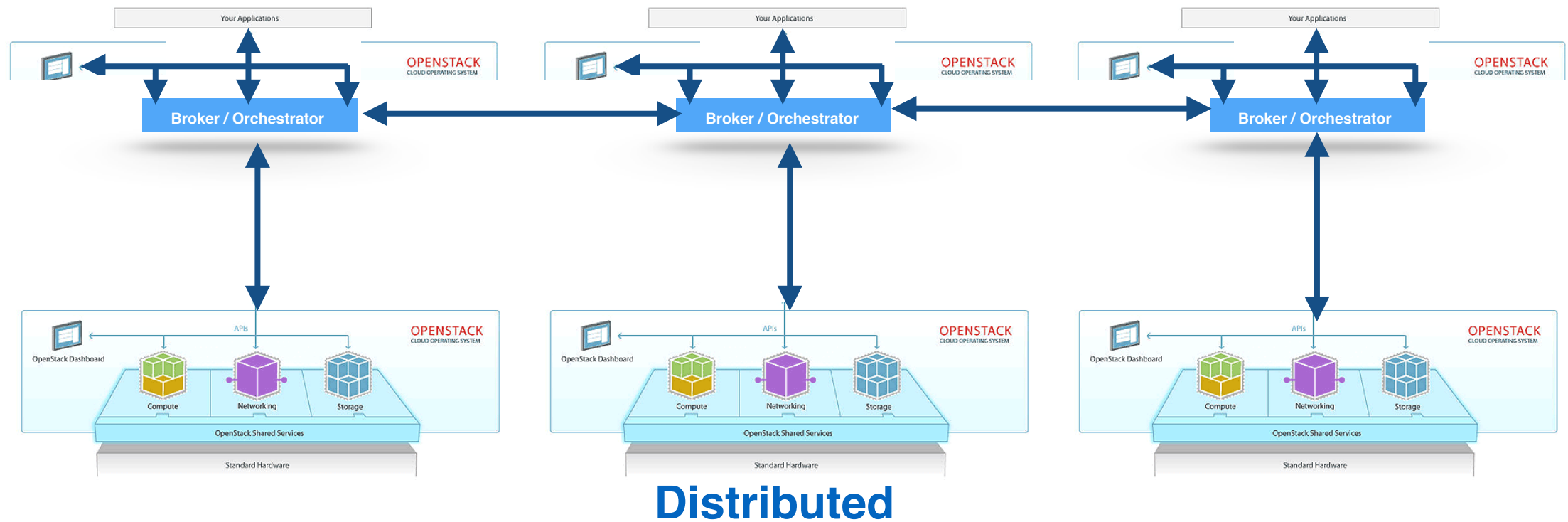
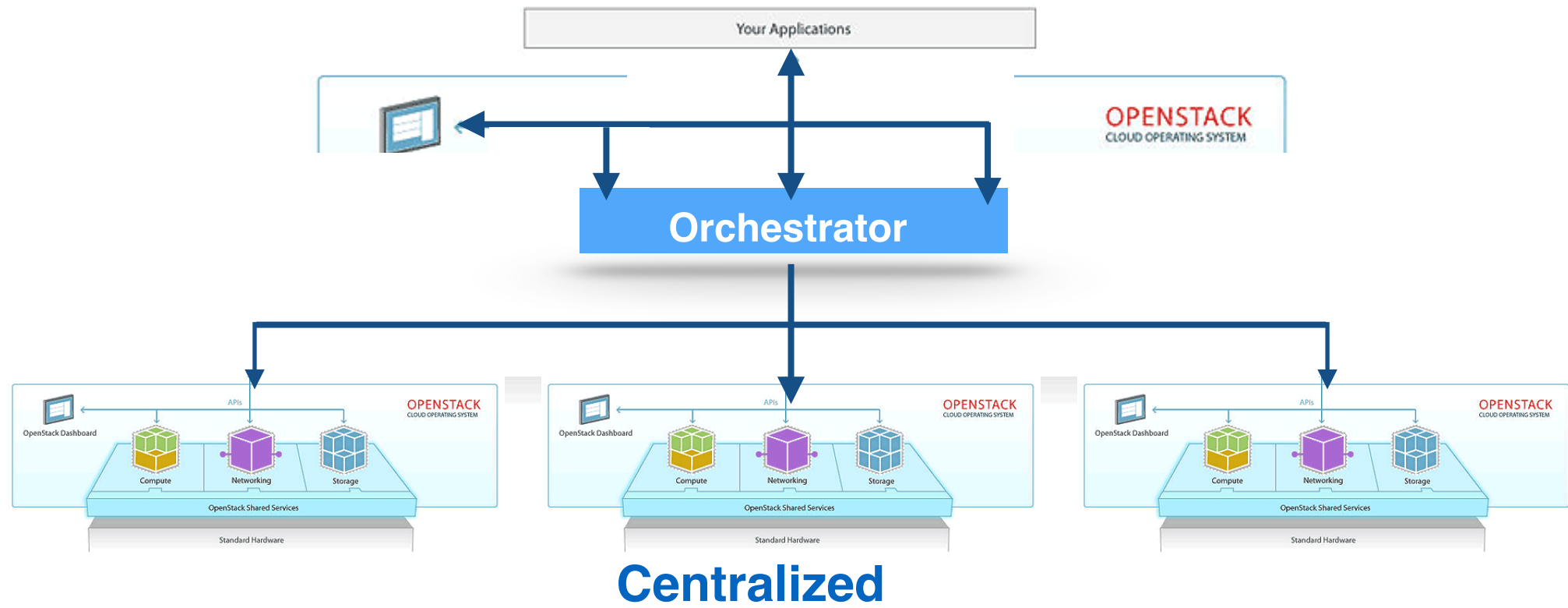
INTERNET2
INTERNET2 NETWORK CONNECTIONS

Development of a fully distributed system in charge of operating such a massively distributed infrastructure

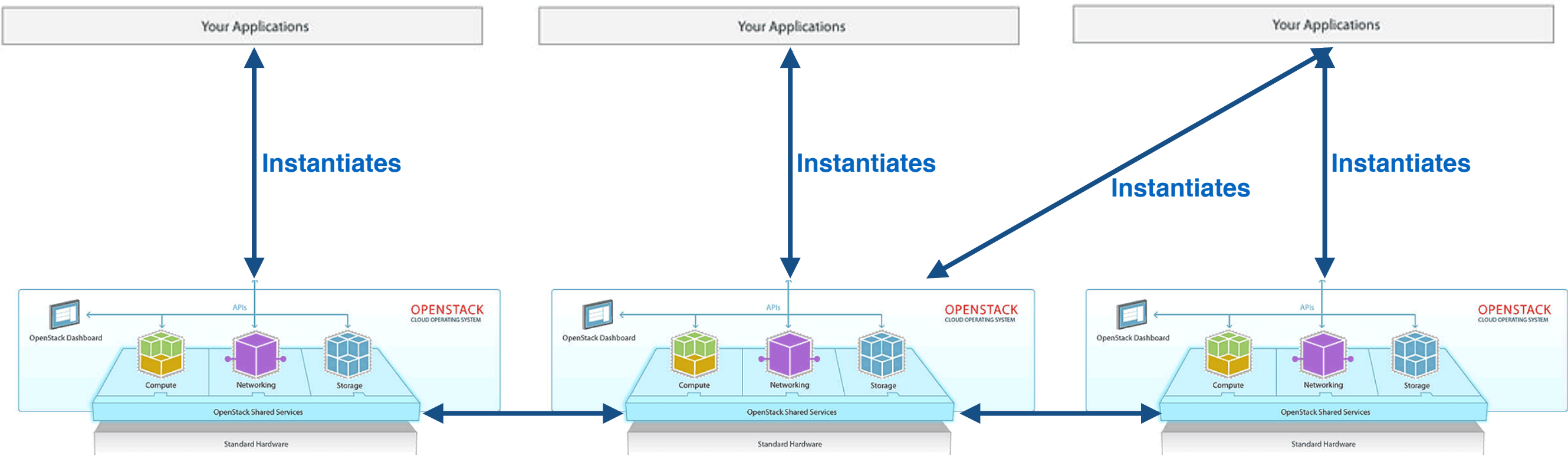


USA NREN

Top-Down



Bottom-Up



Natively distributed

The Discovery Initiative



PoC

(Feasibility investigation)

- Austin Summit
- Redis-based PoC



Performance analysis
(Deep dive into OpenStack)



**Software
prototype**



**Physical
infrastructure**

The Discovery Initiative



PoC

(Feasibility investigation)

- Austin Summit
- Redis-based PoC



Performance analysis (Deep dive into OpenStack)

- Performance WG
- Massively Distributed WG
- Barcelona Summit
- Enos



Software prototype



Physical infrastructure

- System prototype demonstration in an operational environment

The Performance Team

- **Performance Team:** since Mitaka summit
- Part of the Large Deployment Team
 - **Defining** the performance testing and benchmarking methodologies
 - **Driving** solutions within OpenStack libraries and projects
 - **Posting** all data to Performance Docs:
<http://docs.openstack.org/developer/performance-docs/>
 - **Sharing** all scripts, runs and tools

The 1,000 nodes experiment

In conjunction with Mirantis: 2 slightly different approaches to the same experiment



- **1,000 nodes = 1,000 compute nodes**
- Control plane speed/latencies/limits evaluation on scale
- Underlying core-services evaluation (MariaDB, RabbitMQ) for scale
- Study of
 - the services resource consumption
 - potential bottlenecks
 - key configuration parameters
 - the influence of services topologies

1,000 nodes: experimental methodology

Deployment and Benchmark/Monitoring and Analysis tools

- Containers
 - Simplify, monitoring, CI/CD
 - Granularize services/dependencies
 - Flexible placement
 - Simplifies orchestration
- cAdvisor + collectd / InfluxDB / Grafana
- Rally Benchmarks (**boot-and-list** instances scenario)
 - 20,000 VMs
 - Concurrency = 50

1,000 nodes: environment



- Mesos + Docker + Marathon as a platform for Openstack (**15 nodes** with 2xE5-2680, 256GB RAM, 960GB SSD)
- Containerized OpenStack services (Liberty release)
- Modified nova-compute libvirt driver to skip run of qemu-kvm

- **~30 nodes** with powerededge 2xE5-2630, 128GB RAM, 200GB SSD + 3TB HDD (Grid'5000)
- Containerized OpenStack services (Mitaka release)
- **Enos**: augmented Kolla tool
- Use of fake drivers
- Code available:
<https://github.com/BeyondTheClouds/enos>



<> Code

! Issues 9

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

⚡ Pulse

📊 Graphs

⚙ Settings

Experimental eNvironment for OpenStack 🐵 — Edit

🕒 179 commits

🔗 2 branches

🏷 0 releases

👤 5 contributors

📜 GPL-3.0

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



rcherrureau committed on GitHub Merge pull request #26 from msimonin/fix-networks ...

Latest commit a7df40b 4 hours ago

📁 enos	Rewrite init phase using command line interface.	20 hours ago
📁 inventories	Kolla v2 -> v3	a month ago
📁 rally	Fixes BeyondTheClouds/Wiki#37	2 months ago
📁 workload	Change the way benchmarks are launched	12 days ago
📄 .gitignore	Enos refactoring	18 days ago
📄 LICENSE	Update License to GPL-3.0	10 days ago
📄 MANIFEST.in	Enos refactoring	18 days ago
📄 README.md	Update License to GPL-3.0	10 days ago
📄 requirements.txt	Enos refactoring	18 days ago
📄 reservation.yaml.sample	Enos refactoring	18 days ago
📄 setup.py	Rewrite init phase using command line interface.	20 hours ago
📄 tox.ini	Enos refactoring	18 days ago

1,000 nodes: experimental process

Phase 1

Empty
OS

Phase 2

OS under load (Rally)

Phase 3

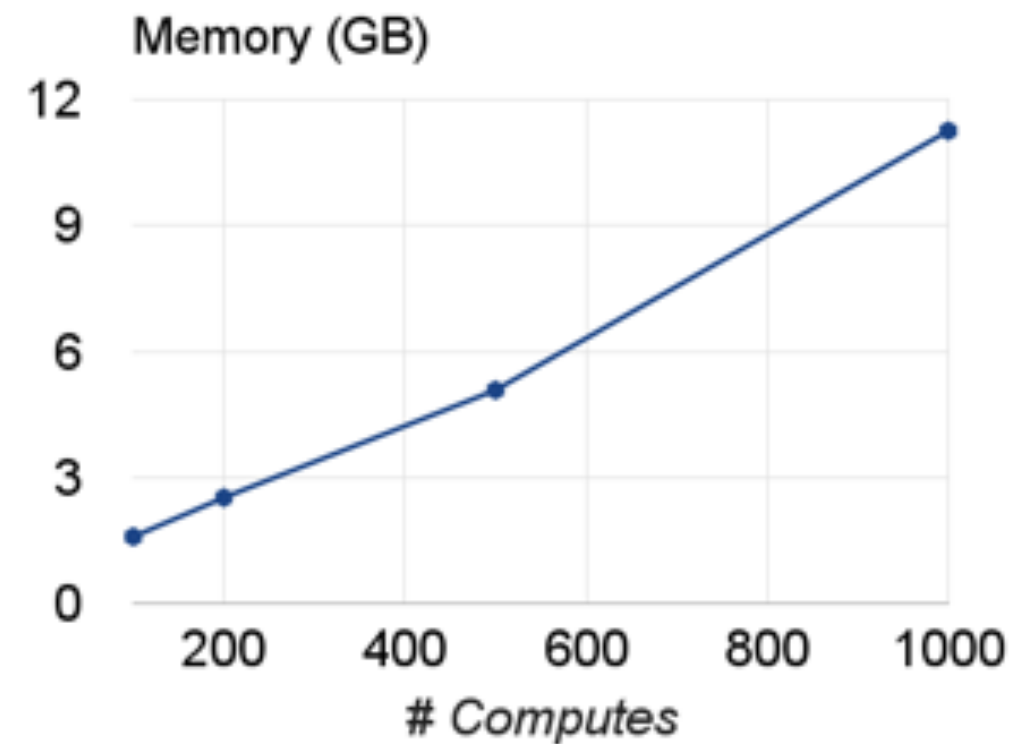
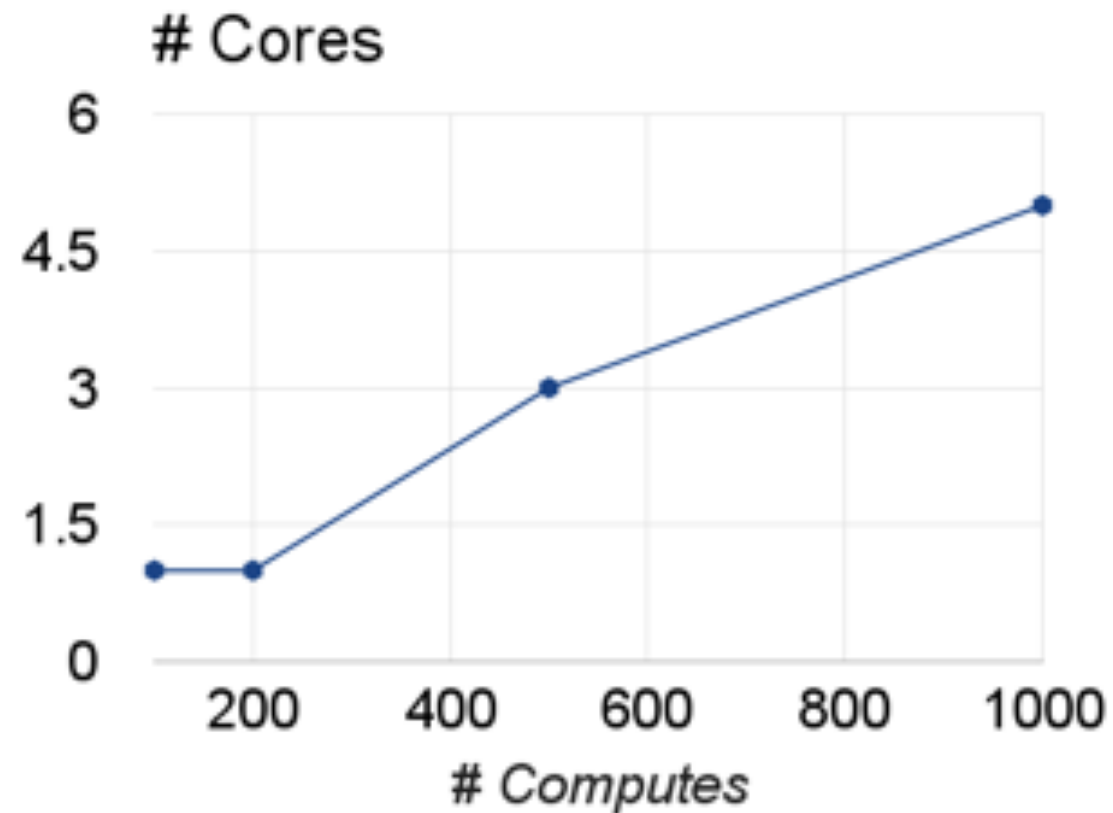
Loaded / idle OS



Boot and List

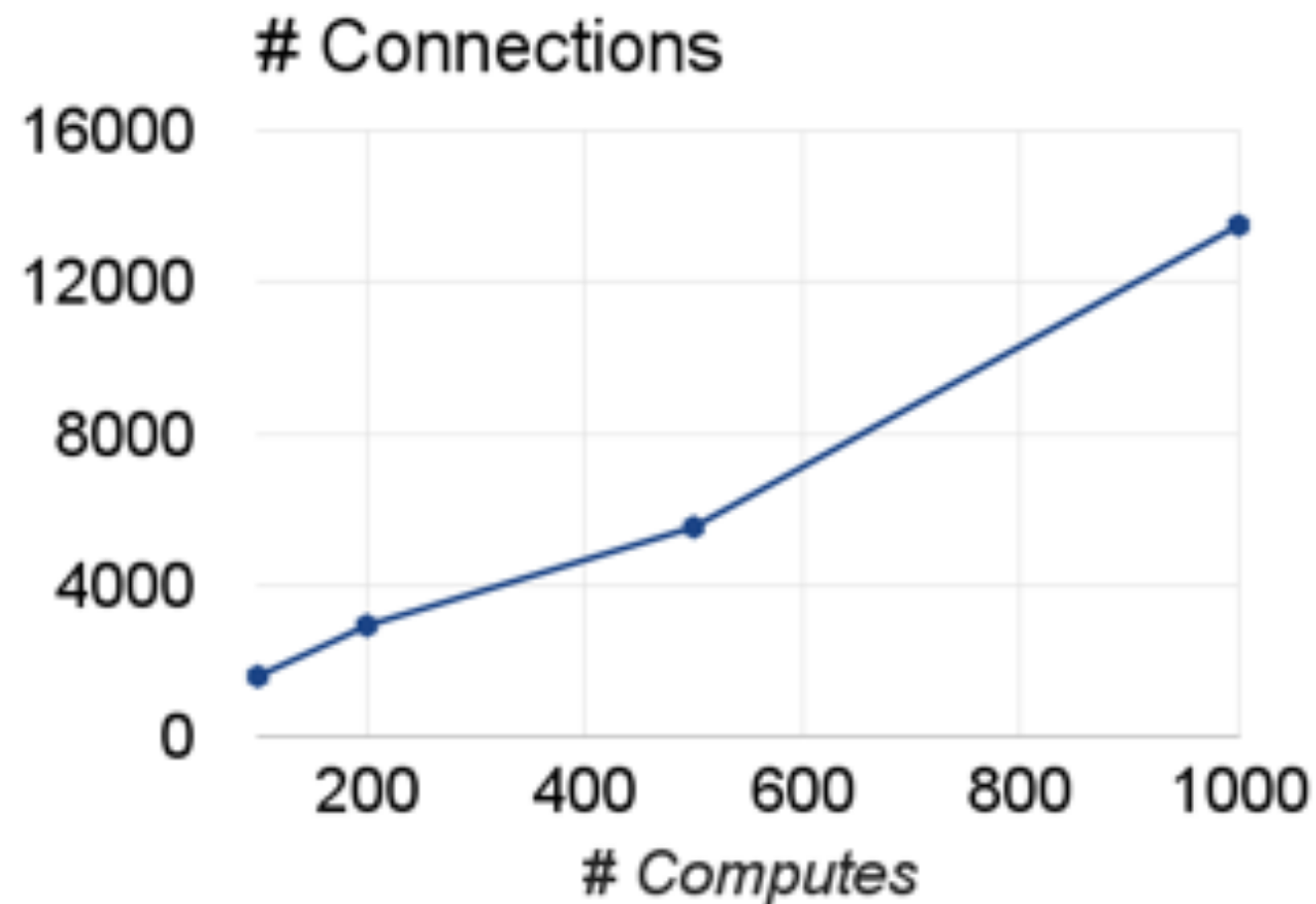
Iterations = 20,000 / Concurrency = 50

1,000 nodes: RabbitMQ (Empty OS)



1 000 nodes : RabbitMQ (Empty OS)

Phase 1



- CPU / RAM / Connections
- Increase linearly with # Computes
- Connections : 15K with 1,000 computes
- RAM: 12 GB with 1,000 computes

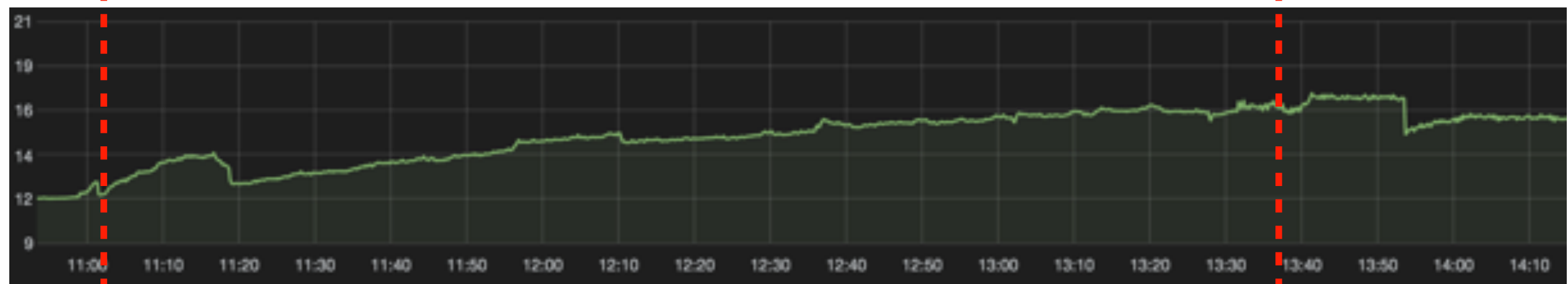
1 000 nodes: RabbitMQ (OS under load)

- (Phase 2) heavy RabbitMQ load, but tolerable, 20 Cores, 17 GB RAM
- (Phase 3) Idle load/Periodic tasks, 3-4 Cores, 16GB RAM.

C
O
R
E
S



R
A
M



1,000 nodes: database (Empty OS)

Database footprints are small even for 1,000 computes

- 0.2 cores
- 600 MB RAM
- 170 opened connections

Effect of periodic tasks for 1,000 computes

- 500 selects / second
- 150 updates / second

1,000 nodes: database (OS under load)

- Database (single node) behaves correctly under load

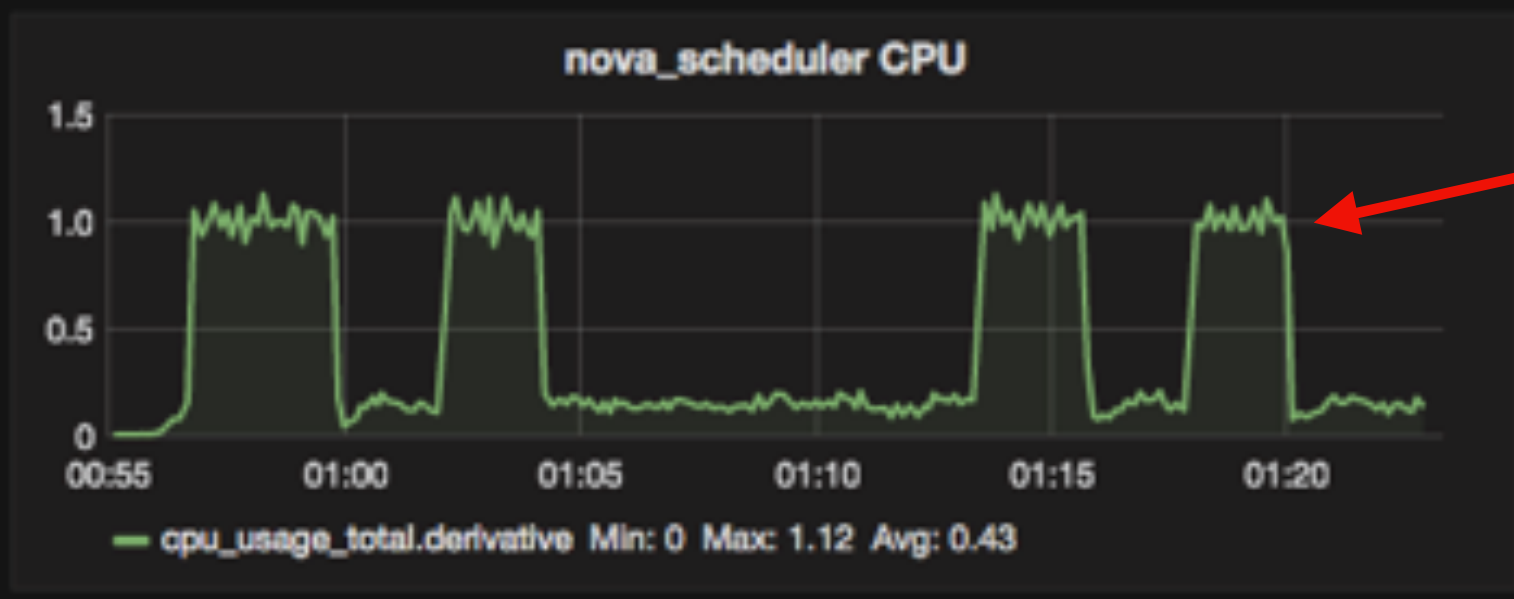


1,000 nodes: nova-scheduler (OS under load)

Rally benchmarks



Nova API: n workers



Scheduler: 1 worker only

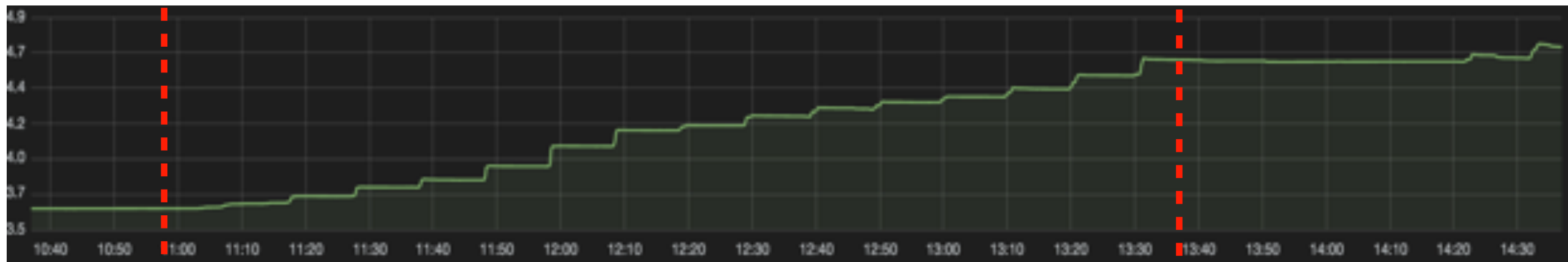
1,000 nodes: nova-conductor (OS under load)

- One of the most loaded services
- Periodic task could be pretty CPU-hungry (up to 30 cores)
- There is no idle time for conductor, unless cloud is empty

C
O
R
E
S



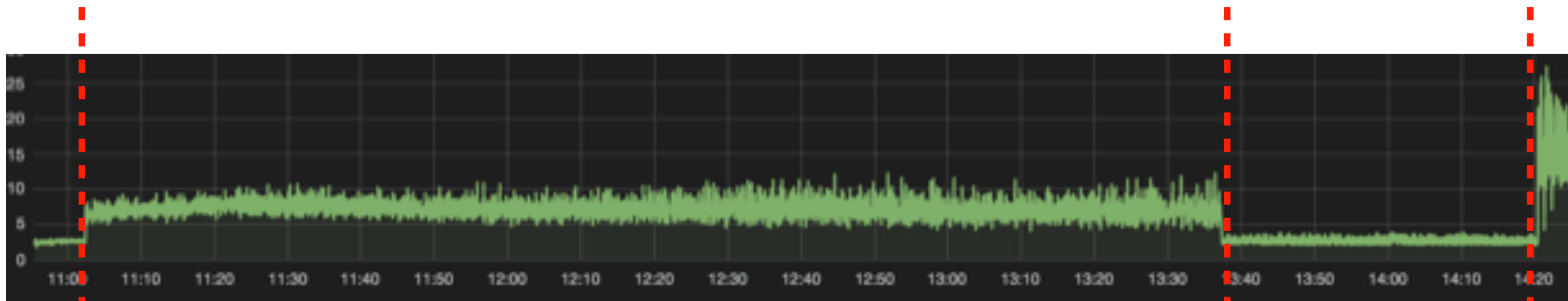
R
A
M



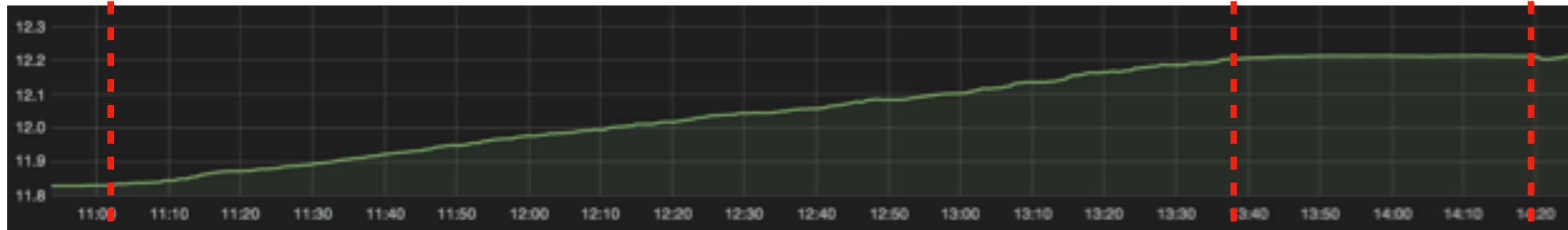
1,000 nodes: nova-api

- Under test load, consumes ~10 Cores; under critical load ~25 Cores
- Without load/Periodic tasks ~3-4 Cores
- RAM consumption is around 12-13GB

C
O
R
E
S



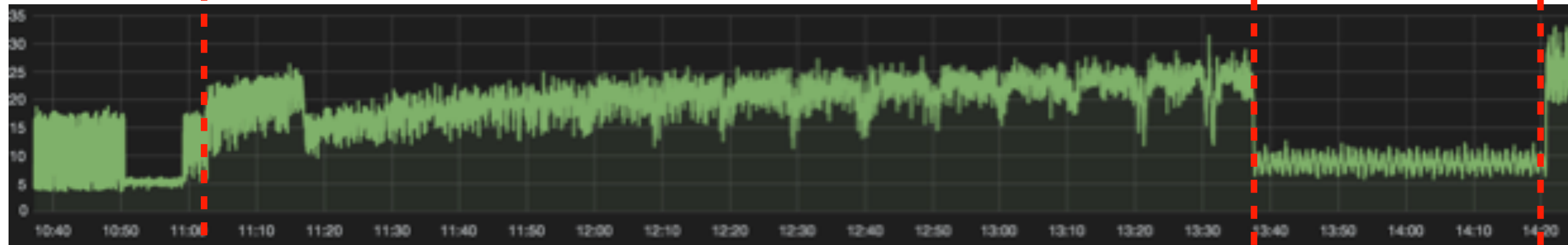
R
A
M



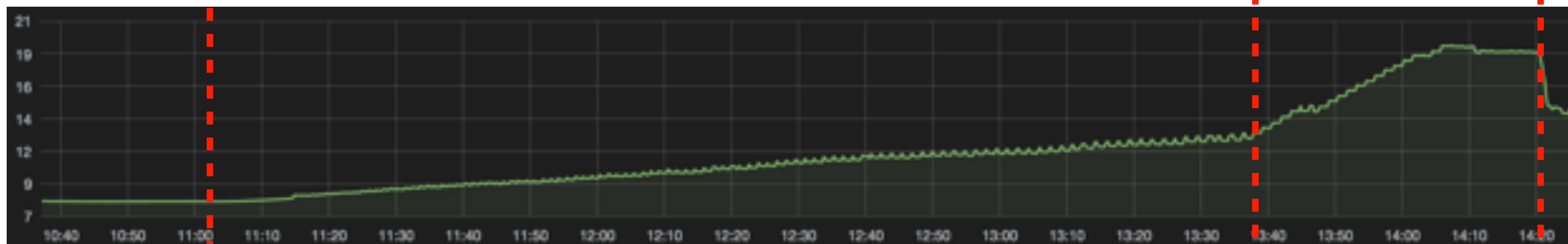
1,000 nodes: neutron-server (API/RPC)

- Under test load consumption is ~30 Cores, under critical ~35 Cores
- Just adding new nodes ~20 Cores, Periodic tasks ~10-12 Cores

C
O
R
E
S



R
A
M



Conclusion

- Default number of API/RPC workers in OpenStack services wouldn't work for us if it tightened up to number of cores.
- MySQL and RabbitMQ aren't a bottleneck at all. At least in terms of CPU/RAM usage. Clustered one's is an additional topic.
- Scheduler performance/scalability issues.

Useful Links

- 1,000 nodes testing:
http://docs.openstack.org/developer/performance-docs/test_plans/1000_nodes/plan.html#reports
- Performance Working Group
 - Team info: https://wiki.openstack.org/wiki/Performance_Team
 - Performance docs:
<http://docs.openstack.org/developer/performance-docs/>
- Weekly meetings at 15:30 UTC, Tuesdays, **#openstack-performance IRC channel:** <https://wiki.openstack.org/wiki/Meetings/Performance>
- And Wednesdays at 15:00 UTC (odd weeks), **#openstack-meeting:** https://wiki.openstack.org/wiki/Massively_Distributed_Clouds